ROAD ACCIDENT INFORMATION SYSTEM DEVELOPMENT OF BATU PAHAT AREA

Mohd Ezree Abdullah[1], Abdul Aziz Chik[2], Mohd Rosli Hainin[2], Ahmad Raqib Ab Ghani[1], Kamaruddin Ambak[1] and Nur Izzi Md Yusoff[3]

ABSTRACT

Batu Pahat is categorised among the highest accident rates in Malaysia. Therefore, excellent accident data management is needed for assisting the diagnosis process of the black spot area. This reason leads to the development of Road Accident Information System (RAIS). The main function of RAIS is to store and manage the accident data efficiently. In this study, the potential of using this system to analyse accident were conducted only in Batu Pahat area. The objective is to develop a database and information system for road accident data. In order to achieve this, Microsoft Access 2003 was utilised incorporating with Microsoft Visual Basic 6.0. As the results, an effective computerised road accident information system had been developed. The system has been calibrated and validated to obtain reliable results.

Keywords: Accident Data Management, Road Accident Information System

1. INTRODUCTION

Traffic accident exacts terrible toll in Malaysia and others countries. The collection and analysis of data on traffic accidents are fundamental to design of programs to reduce that toll. Analysts use accident data to help understand why accidents occur, to help identify accident-prone locations, to aid in deciding which safety programs or countermeasures should be implemented, and to assist evaluations of the countermeasure effectiveness (Hummer 1994).

An essential element of any accident reduction and prevention strategy is the collection and investigation of road accident data. Accident investigation procedures normally depend on the existence of a reliable database (Sabra et al. 2000).

Now, Malaysia is established the database with the microcomputer system that can be used for analyzing this data. But the existing database (software) is not friendly when the researcher, traffic engineer and the person who involve in the road accident analysing process wants to achieve the needed information. Therefore the Road Accident Information System (RAIS) is developed to simplify the management information process. Basically the format of accident report interface is based on the traffic accident report form, POL27 (Pin.1/91).

[1] Faculty of Civil and Environmental Engineering, Kolej Universiti Teknologi Tun Hussein Onn, Parit Raja, Malaysia.
[2] Faculty of Civil Engineering, Universiti Teknologi Malaysia, Skudai, Malaysia.
[3] Faculty of Engineering, Universiti Kebangsaan Malaysia, Bangi, Malaysia

2. METHODOLOGY

The system development methodology is a collection of procedures, techniques, tools and aids documentation which helps the system developers to implement a new information system. A methodology consists of phases, which consists of sub-phases, which will guide the system developers in their choice of techniques that might be appropriate at each stage of the project. It will help them to plan, manage, control, and evaluate information system development projects. Most methodologies are top-down and hierarchical, development efforts will concentrate on defining how high-level processes are inter-related before proceeding with defining low-level detail of processes. System Development Life Cycle (SDLC) and its phases are often supported by the use of Computer Aided Software Engineering (CASE) tools (Russell 2002). These tools support standardization of a development methodology or centralized data for information sharing or check the validity of models.

2.1 Waterfall Model

The waterfall model is a popular version of the systems development life cycle model for software engineering. Often considered the classic approach to the systems development life cycle, the waterfall model describes a development method that is linear and sequential. Waterfall development has distinct goals for each phase of development. Imagine a waterfall on the cliff of a steep mountain. Once the water has flowed over the edge of the cliff and has begun its journey down the side of the mountain, it cannot turn back. It is the same with waterfall development. Once a phase of development is completed, the development proceeds to the next phase and there is no turning back (Kendall K.E and Kendall J.E. 2002).

The advantage of waterfall development is that it allows for departmentalisation and managerial control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process like a car in a carwash, and theoretically, be delivered on time. Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order, without any overlapping or iterative steps. Figure 1 is the classic Waterfall model methodology, which is the first SDLC method and it describes the various phases involved in development of RAIS.

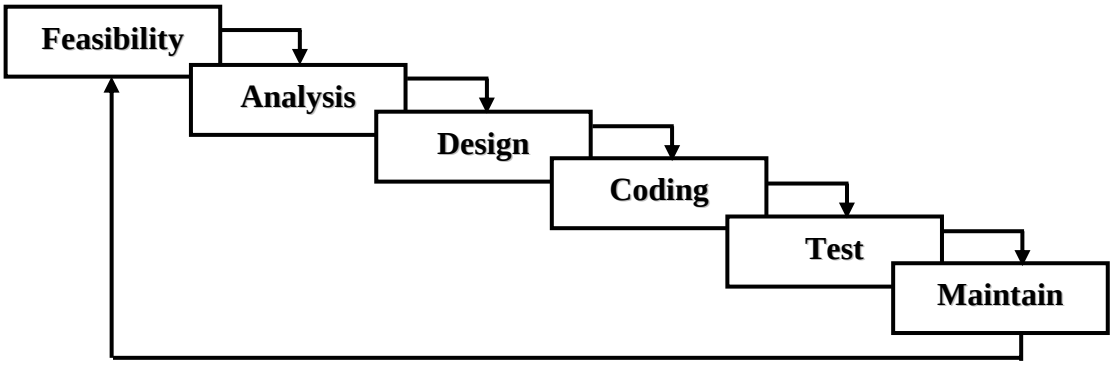


FIGURE 1: The classic Waterfall Model

Phase I: Feasibility and System Planning

In this phase of the systems development life cycle, the analyst was concerned with identifying problems, opportunities, and objectives. The stage was critical to the rest of the project, because no one wants to waste subsequent time addressing the wrong problem. Identifying the objectives was also an important component to do in the first phase.

Phase II: Requirement Analysis

Analysis gathers the requirements for the system. This stage includes a detailed study of the needs of the organization. Design focuses on high level design like, what programs are needed and how are they going to interact, low-level design (how the individual programs are going to work), interface design (what are the interfaces going to look like) and data design (what data will be required). During these phases, the software's overall structure is defined. Analysis and Design are very crucial in the whole development cycle. Any glitch in the design phase could be very expensive to solve in the later stage of the software development. Much care is taken during this phase. The logical system of the product is developed in this phase.

The main output of this phase is software specifications, which was the detail statement of the system function in order to achieve the objectives. Figure 2 shows the requirement analysis process.

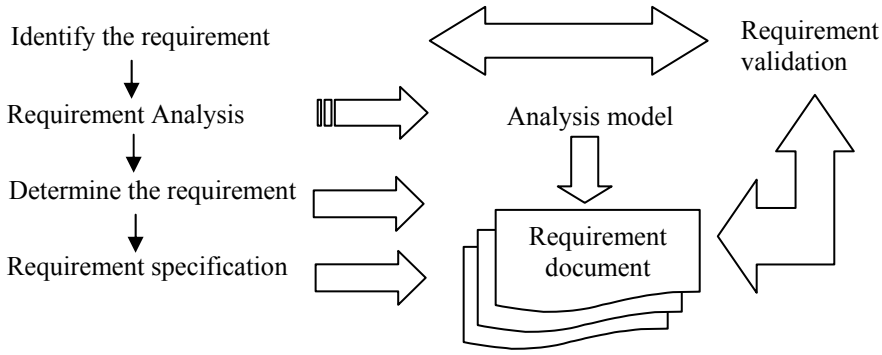


FIGURE 2: Requirement analysis process

a. Identify the Requirement

In order to get the information sources about the system requirement, several approaches are used to get user requirement as stated below such as interview, observation, document study and discussion.

b. Requirement Analysis

In this phase special tools and techniques help to make requirement determinations. Such tool used was the data flow diagram (DFD) to chart the input, process and the output of the system. The other tool used in this phase is the entity relationship (ER) diagram. ER is a way of graphically representing the logical relationships of entities.

During this phase, the system analyst also analyzes the structured decisions made. Structured decisions were those for which the conditions, condition alternatives, actions, and action rules had been determined.

The output of this phase must be presented and documented in the simple way. Analysis approach is based on programming concept which called object oriented. Generally, this type of

system model is assumed as the abstract of the developed system. Therefore, all the entities involve in the system must be presented in a model form.

i. Entity Relationship Diagram (ERD)

The entity-relationship model is a way of graphically representing the logical relationships of entities (or objects) in order to create a database. In ER modelling, the structure for a database is portrayed as a diagram, called an entity-relationship diagram (or ER diagram), that resembles the graphical breakdown of a sentence into its grammatical parts. Entities are rendered as points, polygons, circles, or ovals. Relationships are portrayed as lines connecting the points, polygons, circles, or ovals. Any ER diagram has an equivalent relational table, and any relational table has an equivalent ER diagram. ER diagramming is an invaluable aid to engineers in the design, optimization, and debugging of database programs.

In a logical sense, entities are the equivalent of grammatical nouns, such as employees, departments, products, or networks. An entity can be defined by means of its properties, called attributes. Relationships are the equivalent of verbs or associations, such as the act of purchasing, the act of repairing, being a member of a group, or being a supervisor of a department. A relationship can be defined according to the number of entities associated with it, known as the degree while the number of the relationship is called cardinality. Figure 3 shows the entity relationship diagram for RAIS system which consists of three entities.

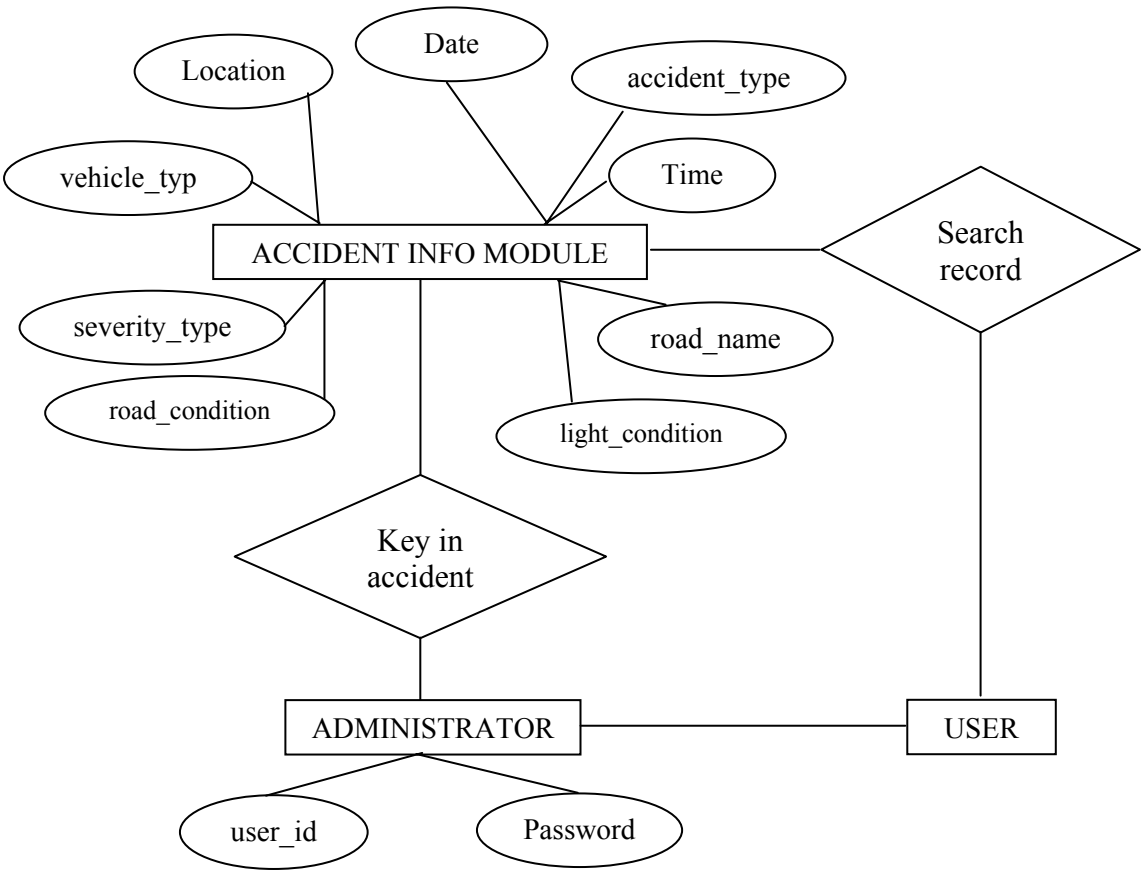


FIGURE 3: ERD for RAIS system

ii. Data Flow Diagram (DFD)

When system analysts attempt to understand the information requirements of users, they must be able to conceptualise how data move through the organisation, the processes or transformation that the data undergo, and what the outputs are. Although interviews and the investigation of hard data provide a verbal narrative of the system, a visual depiction can crystallise this information in useful ways.

Through a structured analysis technique called data flow diagram (DFD), the system analyst can put together a graphical representation of data processes throughout the organisation. DFD focuses on the data flowing into and out of the system and the processing of data. These basic components of every computer program may be described in detail and used to analyse the system for accuracy and completeness.

The data flow approach emphasizes the logic underlying system. By using combination of only four symbols, the system analyst can create a pictorial depiction of processes that will eventually provide solid system documentation. Figure 4 to Figure 6 shows the DFD for RAIS System.

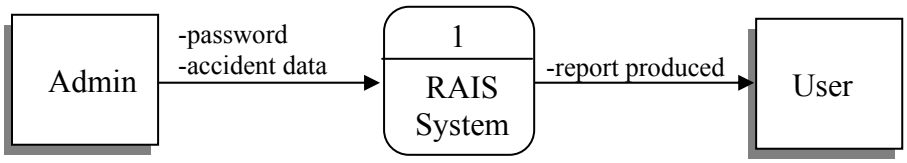


FIGURE 4: DFD level 0

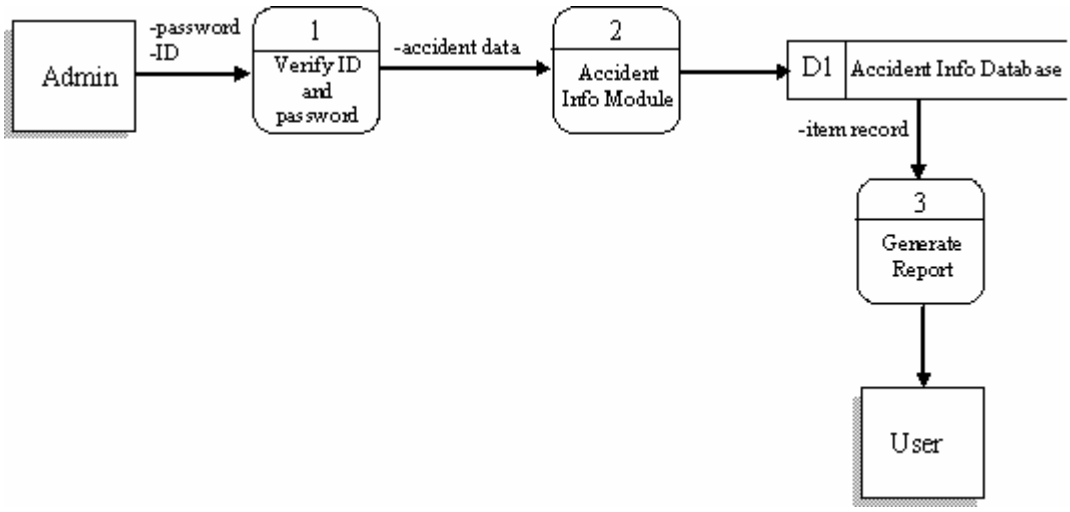


FIGURE 5: DFD level 1

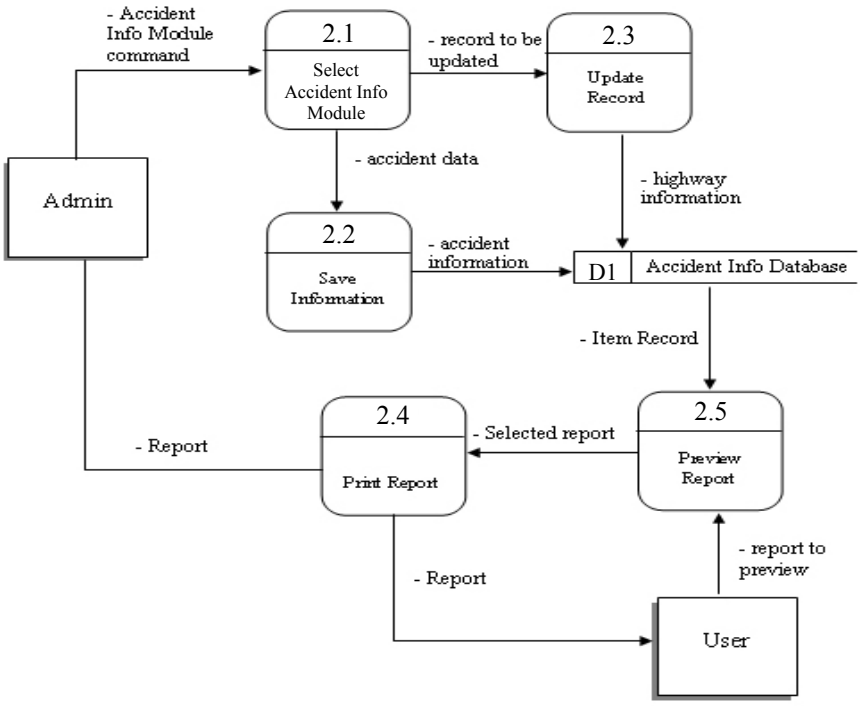


FIGURE 6: DFD level 2

iii. Determination of Requirement

The next phase that the analyst enters is that of determining information requirements for the particular users involved. Among the tools used to define information requirements in the organisations are sampling and investigating hard data, interviewing, questionnaires, observing decision maker's behaviour and even prototyping. In this phase, the details of current system function are needed.

iv. Requirement Specification

Requirement specification helps to analyse the requirement in details, so the specification and the determination is synchronised. Basically, requirement specification is presented in System Model that developed in requirement analysis process.

Phase III: System Design

In the design phase of the systems development life cycle, the information collected earlier is used to accomplish the logical design of the information system. The accurate data entries are designs so that data going into the information system are correct. In addition, the effective input is provided to the information system by using techniques of good form and screen design (Kendall K.E and Kendall J.E. 2002).

Part of the logical design of the information system is devising the user interface. The interface connects the user with the system and is thus extremely important. Examples of user interface include a keyboard (to type in questions and answer), onscreen menus (to elicit user commands), and a variety of Graphical User Interface (GUI) that use a mouse or touch screen.

The design phase also includes designing files or databases that will store much of data needed by decision makers in the organization. A well organized database is the basis for all information systems. Finally, the control and backup procedures are designs to protect the system and the data and to produce program specification packets for programmers.

Phase IV: Coding

The design must be translated into a machine-readable form. The code generation step performs this task. If the design is performed in a detailed manner, code generation can be accomplished without much complication (Perry and Hettihewa 1998). Programming tools like compilers, interpreters, debuggers are used to generate the code. Different high level programming languages like C, C++, Pascal and Java are used for coding. With respect to the type of application, VB6 had been chosen. The criteria of choosing the programming language were:

i. Application field that relate to the use of database
ii. Algorithm that is used
iii. Skill and experience of end user
iv. The ability of the system to detect error
v. Environment of the system implementation

Phase V: Program Testing

After code generation phase the software program testing begins. Different testing methods are available to detect the bugs that were committed during the previous phases. A number of testing tools and methods are already available for testing purpose.
Some organisations build their own testing tools that are tailor made for their own development operations. The methods of the program testing that used in this research are unit testing, integration testing, system testing and acceptance testing.

Phase VI: Operation and Maintenance

Software will definitely go through change once when it is delivered to the customer. There are large numbers of reasons for the change. Change could happen due to some unpredicted input values into the system. In addition to this the changes in the system directly have an effect on the software operations. The software should be implemented to accommodate changes that could be happen during the post development period.

Maintenance is performed for two reasons. The first of these is to correct software errors. No matter how thoroughly the system is tested, bugs or errors creep into computer programs. The other reason for performing system maintenance is to enhance the software's capabilities in response to changing organisational needs.

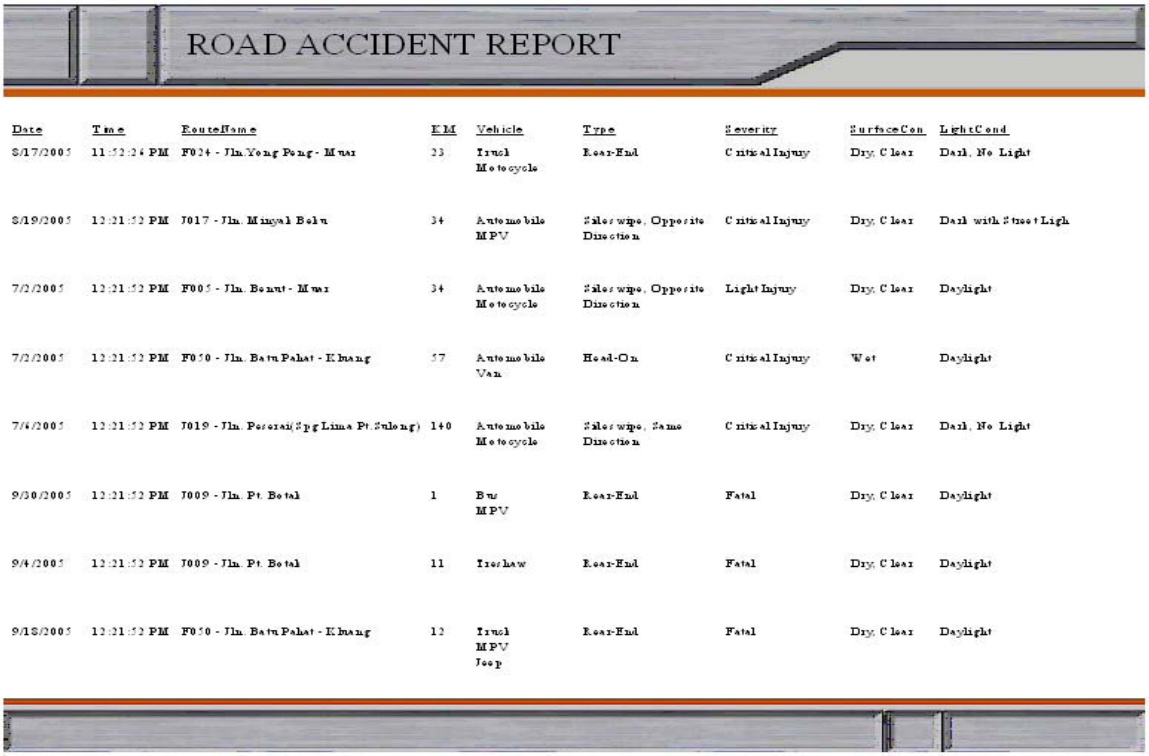
3. SYSTEM DISPLAY AND DISCUSSION

3.1 Introduction

The development of Road Accident Information System (RAIS) was to help the administrator at the Public Works Department (PWD) of Batu Pahat in managing the road accident information efficiently. Besides that, RAIS was developed to facilitate the traffic engineer, researcher, students and etc to perform the accident analysis effectively and also to obtain the needed information easily.

Figure 7 shows the 'Road Accident Information' interface. In this section, accident details such as date and time of accident, route name, location, type of vehicle, accident and severity, road surface condition and lighting condition are stored in accident information database. All the information stored can be edit by clicking the 'Find to edit' button. Administrator can also view all the record in a report form by clicking the 'Report' button (Figure 8). The report can be print out by clicking the print button.

FIGURE 7: The Road Accident Information interface



Date	Time	RouteName	KM	Vehicle	Type	Severity	SurfaceCon	LightCond
8/17/2005	11:52:24 PM	F024 - Jln.Yong Peng - Muar	23	Truck Motocycle	Rear-End	Critical Injury	Dry, Clear	Dark, No Light
8/19/2005	12:21:52 PM	J017 - Jln. Mingyal Bolu	34	Automobile MPV	Sides wipe, Opposite Direction	Critical Injury	Dry, Clear	Dark with Street Ligh
7/2/2005	12:21:52 PM	F005 - Jln. Benut - Muar	34	Automobile Motocycle	Sides wipe, Opposite Direction	Light Injury	Dry, Clear	Daylight
7/2/2005	12:21:52 PM	F050 - Jln. Batu Pahat - Kluang	57	Automobile Van	Head-On	Critical Injury	Wet	Daylight
7/4/2005	12:21:52 PM	J019 - Jln. Pesrai(Jpg Lima Pt.Sulong)	140	Automobile Motocycle	Sides wipe, Same Direction	Critical Injury	Dry, Clear	Dark, No Light
9/30/2005	12:21:52 PM	J009 - Jln. Pt. Betal	1	Bus MPV	Rear-End	Fatal	Dry, Clear	Daylight
9/4/2005	12:21:52 PM	J009 - Jln. Pt. Betal	11	Trechaw	Rear-End	Fatal	Dry, Clear	Daylight
9/18/2005	12:21:52 PM	F050 - Jln. Batu Pahat - Kluang	12	Truck MPV Jeep	Rear-End	Fatal	Dry, Clear	Daylight

FIGURE 8: The Road Accident Information report

Besides those buttons on the main menu of the administrator part, there is another button called the 'Map of Batu Pahat' button. The display of Batu Pahat map as shown in Figure 9 will appear when the administrator clicks that button.

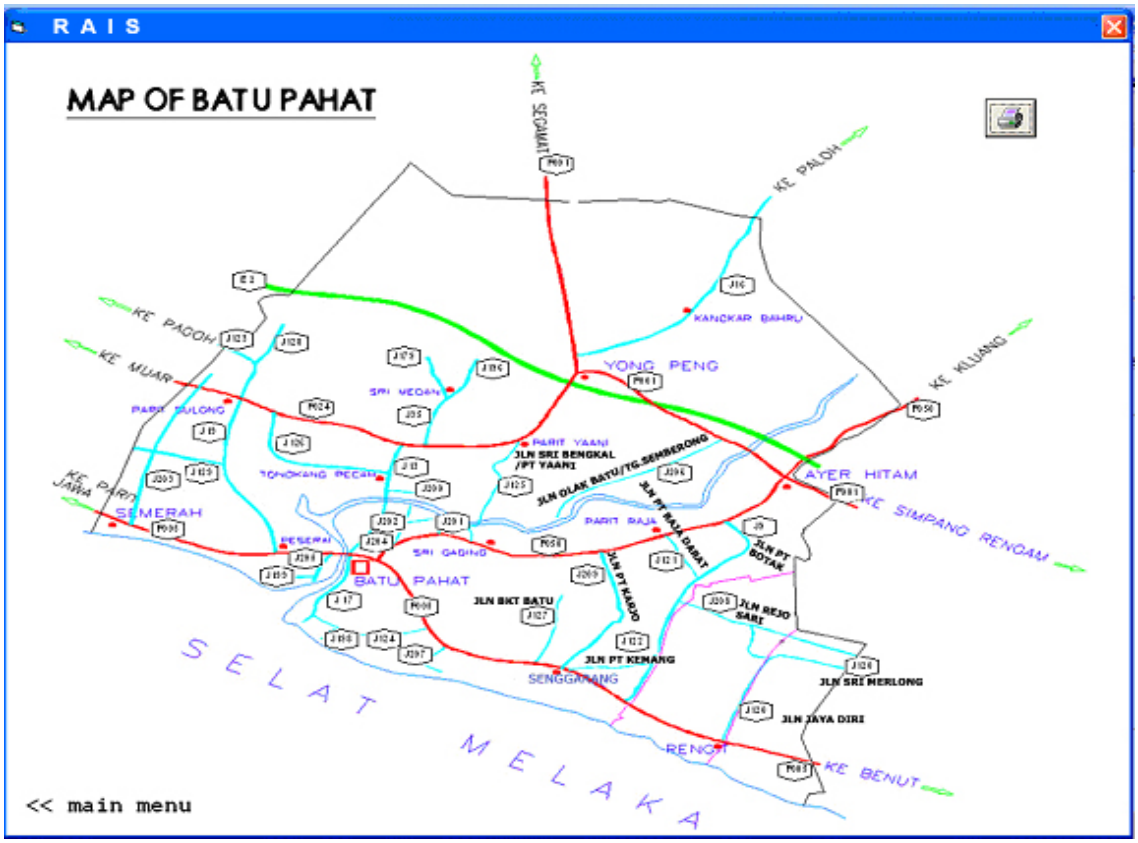


FIGURE 9: Map of Batu Pahat interface

3.2 The User Part

In this part, user can only perform the searching record process while. All the report needed can be printed out by clicking the Show Report button. Figure 10 shows the interface the user part.

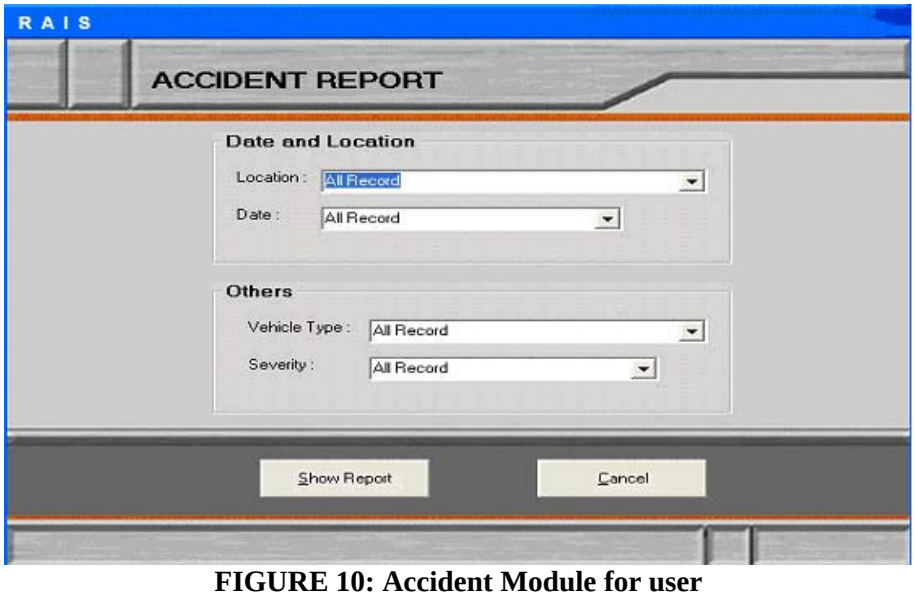


FIGURE 10: Accident Module for user

4. CONCLUSION

Based on the observation, problems and needs have been identified such as time management for information collecting and also data analysing. For time management problem, the information is stored in different sources such as accident information at Police Traffic Department and highway information at Public Work Department (PWD). It is inconvenience if the information is stored at difference sources at the same time.

Therefore, the Road Accident Information System (RAIS) for Batu Pahat district had fulfilled the aim and objectives which were to manage accident information efficiently.

RAIS is a user-friendly system because it has the icons, buttons and menu system that is very common to the user. To prevent the recorded information from being hacked, RAIS has its own security system.

The implementation of RAIS solves the problems of authority at PWD in storing the data efficiently. Besides that, users can save their time and energy in searching the required information because the accident information are now integrated at the same department.

REFERENCE

Hummer, J.E. (1994). "Traffic Accidents Studies." in Robertson, H.D. "*Manual of Transportation Engineering Studies.*" New Jersey: Prentice Hall. 191-218.

Kendall, K.E and Kendall, J.E. (2002). "*System Analysis and Design.*" New Jersey: Prentice Hall International.

Perry, G and Hettihewa, S. (1998). "*Visual Basic 6.0 in 24 Hours.*" Indiana: Sams Publishing.

Polis Diraja Malaysia (1991). "*POL 27 (Pindaan 1/91) – Panduan Mengisi Borang.*" Kuala Lumpur: Cawangan Trafik, Ibu Pejabat Bukit Aman.

Russell, K. (2002). "QuickStudy: System Development Life Cycle." in "*Computerworld.*"

Sabra, Z., Wallace, C.E and Lin, F.B. (2000). "*Traffic Analysis Software Tools.*" National Research Council. Washington D.C.